Routing Exercise

1.

Consider the following instance of channel routing problem:

TOP = [2, 3, 3, 6, 0, 7, 1, 6, 5, 4]

BOT = [0, 1, 3, 2, 7, 3, 7, 4, 1, 5]

where 0 denotes a vacant terminal. Assume two layer channel routing in all cases.

(a)

Calculate the local density at each column. From the local density derive a lower bound on channel width for successful routing.

(b)

Draw the vertical constraint graph (VCG). From the VCG derive a lower bound on channel width for successful routing.

(c)

Draw the vertical constraint graph (HCG). From the HCG derive a lower bound on channel width for successful routing.

(d)

Apply the greedy algorithm to the above instance of channel routing problem. Show the final routing solution obtained.

Solution:

a)

* Channel routing instance can be represented as:

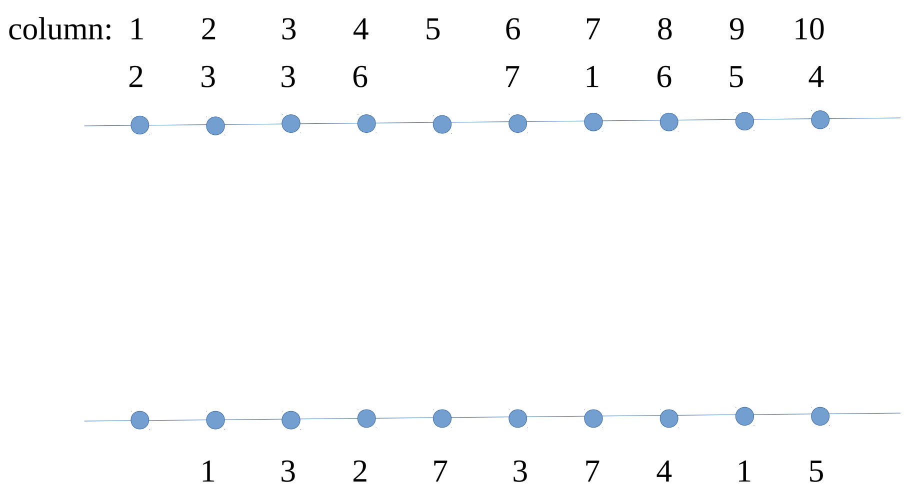

Figure 1: Channel Routing Instance.

* Decomposing each multi-pin net into 2-pin connections:

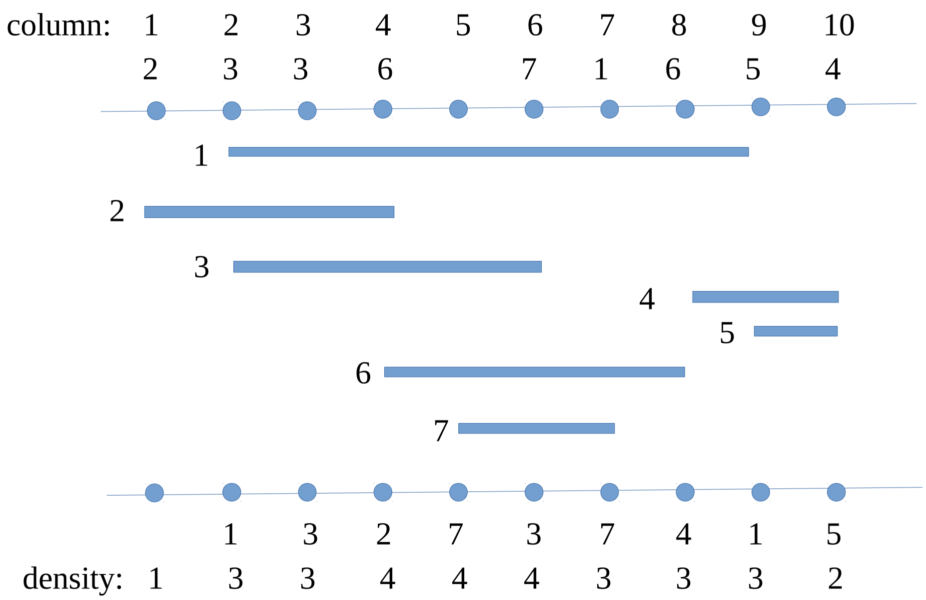

Figure 2: Multi-pin net decomposition

Hence, the local desities for columns 1-10 are: 1, 3, 3, 4, 4, 4, 3, 3, 3, 2
i.e ld(1) = 1, ld(2) = 3, ld(3) = 3, ld(4) = 4, ld(5) = 4
ld(6) = 4, ld(7) = 3, ld(8) = 3, ld(9) = 3, ld(10) = 2

Therefore, the channel density, d = max{ ld(C) } over all C
= 4

i.e. d = 4

---

b)

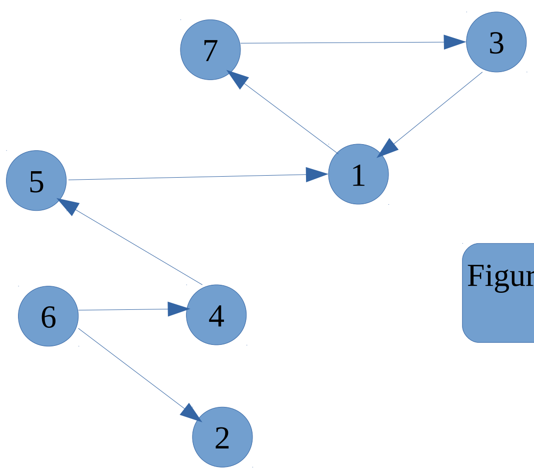

Figure 3: Vertical constraint graph (VCG) for Channel Routing instance.

Lower bound for number of tracks (without doglegs) = Longest path in VCG = 6

---

c)

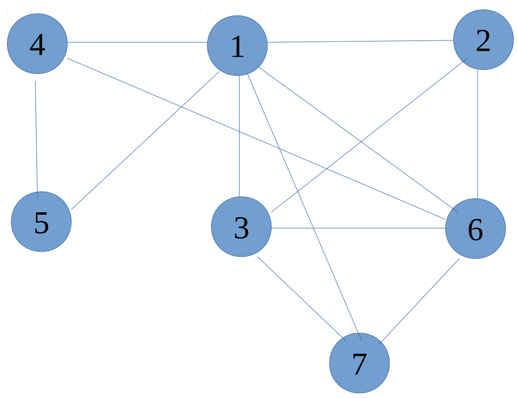

Figure 4: Horizontal constraint graph (HCG) for Channel Routing instance.

Channel density (lower bound on number of tracks needed for routing) = Maximal clique in HCG = 4

---

d) Channel-length, λ = 10

Top connection list, T = (2, 3, 3, 6, 0, 7, 1, 6, 5, 4)
Bottom connection list, B = (0, 1, 3, 2, 7, 3, 7, 4, 1, 5)

Left connection set, L = { 2 }
Right connection set, R = {4, 5}

H(1) = 9, H(2) = 4, H(3) = 6, H(4) = 10, H(5) = 10,
H(6) = 8, H(7) = 7

initial-channel-width = 4
minimum-jog-length = w/4 = 1 (w=4, i.e. best channel width obtainable)
steady-net-constant = 10
In the following diagrams, orange lines indicate jogs.

* **Assign tracks to nets at left end:** => Y(2) = {1}

Column 1:

a) <u>Make Feasible Top and Bottom Connections in Minimal Manner:</u>

Connect T_1 = 2 to track 1 => Y(2) = {1} => No split nets => No need for step (b), (c)

d) <u>Add Jogs to Raise Rising Nets and Lower Falling Nets:</u>

Add jog to to lower falling net 2 to an empty track (track 4) which is as close as possible to its target edge. => Y(2) = {4}

* No need for step (e) i.e. no need to widen channel

f) Extend to Next Column

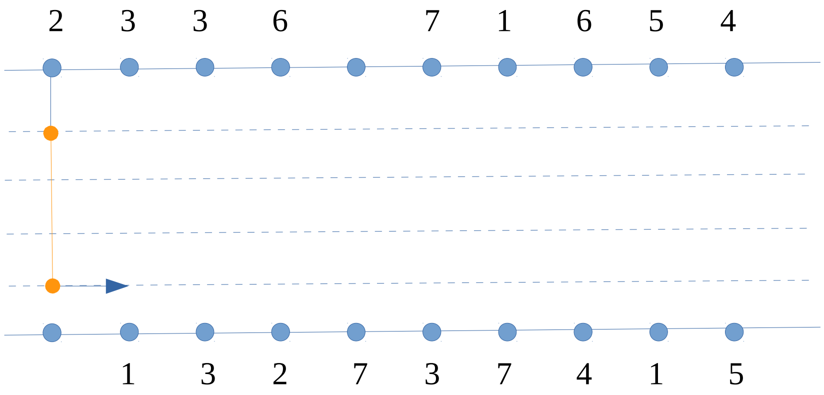

Figure 5: Routing for given routing instance after column 1 has been routed using rules as per greedy channel router algorithm.

Column 2:

a) Make Feasible Top and Bottom Connections in Minimal Manner:

Connect $T_2 = 3$ to track 1 and $B_2 = 1$ to track 3.

=> Y(2) = {4}, Y(3) = {1}, Y(1) = {3} => No split nets => No need for steps (b) and (c)

* No need for step (d) because net 2 is already closest to its target edge and net 3 and net 1 are steady nets.

* No need for step (e) i.e. no need to widen channel

f) Extend to next column

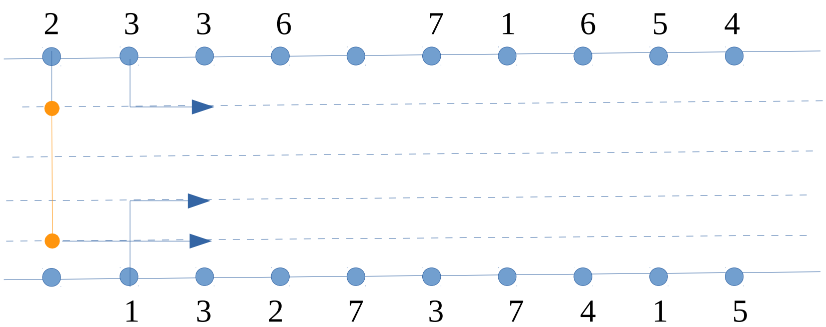

Figure 6 : Routing for given routing instance after column 2 has been routed using rules as per greedy channel router algorithm.

Column 3:

a) Make Feasible Top and Bottom Connections in Minimal Manner:

Connect $T_3 = 3$ to track 1 and $B_3 = 3$ to track 2. => Y (2) = {4}, Y(3) = {1,2}, Y(1) = {3} => | Y(3)| > 1

i.e. net 3 is a split net or a collapsible net.

b) Free Up As Many Tracks As Possible By Collapsing Split Nets:

Added pattern of one “collapsing jog” to connect tracks 1 and 2, both holding net 3. This pattern free's up one track => Y(3) = {2}, Y(2) = {4}, Y(1) = {3}

* No need for step (c) since, no split nets remain.
* No need for step (d) since, because net 2 is already closest to its target edge and net 1 is a steady nets. Jog is already added in step b which brings falling net 3 closest to its target edge.
* No need for step (e) because T_3 and B_3 were brought in to the track.

f) <u>Extend to next column</u>

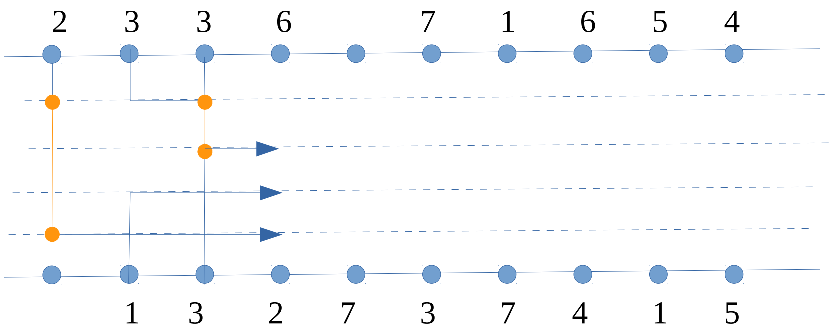

Figure 7: Routing for given routing instance after column 3 has been routed using rules as per greedy channel router algorithm.

Column 4:

a) <u>Make Feasible Top and Bottom Connections in Minimal Manner:</u>

Connect $T_4 = 6$ to track 1 and $B_4 = 2$ to track 4 i.e. track already assigned to this net.

=> Y(2) = {}, Y(3) = {}, Y(1) = {3}, Y(6) = {1} => No split nets => No need for steps (b) and (c)

* No need for step (d) because net 6 is already closest to its target edge and net 1 is a steady net.
* No need for step (e) i.e. no need to widen channel because T_4 and B_4 were brought in to the channel

f) <u>Extend to next column</u>

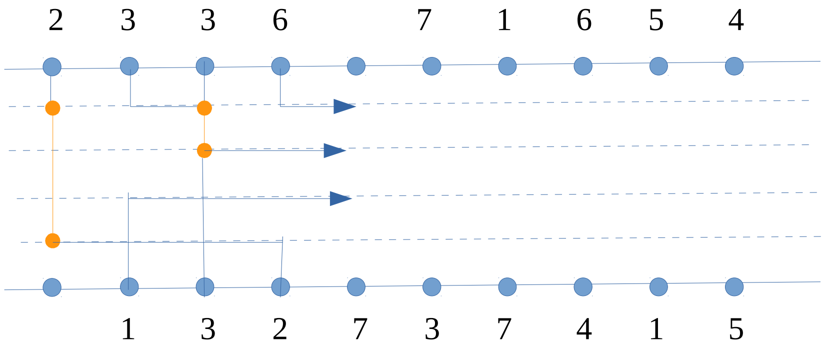

Figure 8: Routing for given routing instance after column 4 has been routed using rules as per greedy channel router algorithm.

Column 5:

a) Make Feasible Top and Bottom Connections in Minimal Manner:

Connect $B_5 = 7$ to track 4 i.e. to the nearest possible track which is empty.

=> Y(2) = {}, Y(3) = {2}, Y(1) = {3}, Y(6) = {1}, Y(7) = {4}

=> No split nets => No need for steps (b) and (c)

* No need for step (d) because net 6 is already closest to its target edge and net 1 and net 3 are steady net. Tracks 1, 2 and 3 are already occupied so net 7 is as close as possible to its target edge.
* No need for step (e) i.e. no need to widen channel because T_5 is a vacant terminal and B_5 was brought in to the channel

f) Extend to next column

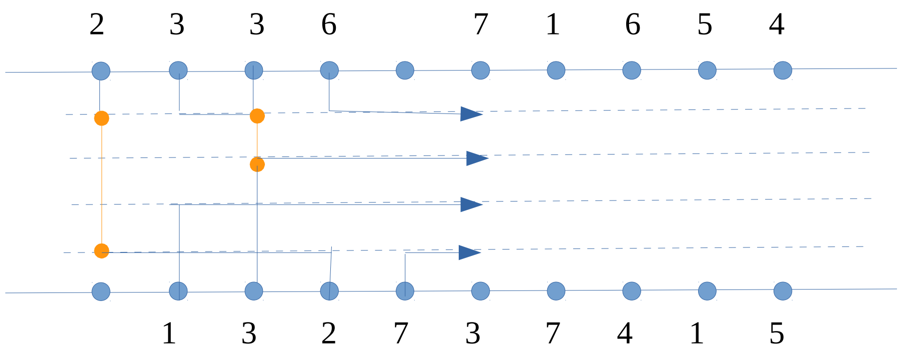

Figure 9: Routing for given routing instance after column 5 has been routed using rules as per greedy channel router algorithm.

Column 6:

a) Make Feasible Top and Bottom Connections in Minimal Manner:

Since, $T_6 \neq B_6$ and the vertical segments from the two terminals would conflict (overlap) then bring in B_6 since it can be brought in with the least wire and we leave the other net to be brought in at step (e). This “finishes” net 3.

=> Y(2) = {}, Y(3) = {}, Y(1) = {3}, Y(6) = {1}, Y{7} = {4}

=> No split nets => No need for steps (b) and (c)

* No need for step (d) because net 6 is already closest to its target edge and net 1 is a steady net. Tracks 1, 2 and 3 are already occupied so net 7 is as close as possible to its target edge.

e) Widen Channel If Needed To Make Previously Infeasible Top Or Bottom Connection:

Since, $T_6 = 7$ could not be brought in to a track in step (a), we create a new track for this net and bring the net in to this track. The track is placed to be track 2 so that it is as near the center of the channel as possible between existing tracks. Modify the number of the tracks after this step.

Also, the Y(n) values are updated as:

=>Y(6) = 1, Y(7) = {2, 5}, Y(3) = {}, Y(1) = 3, Y(2) = {}

f) Extend to next column

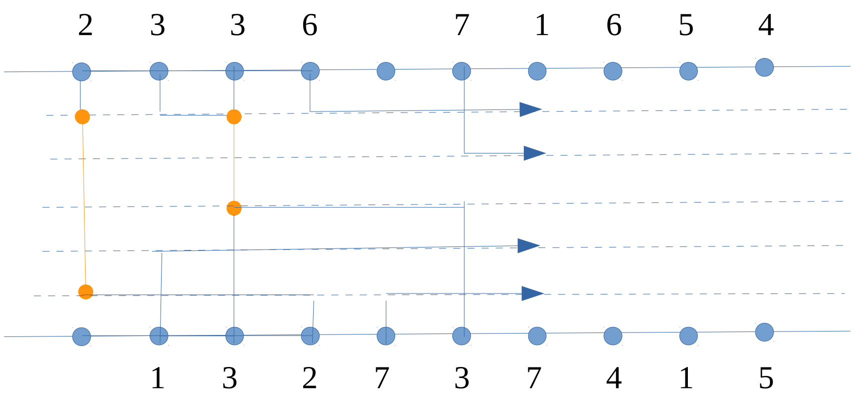

Figure 10: Routing for given routing instance after column 6 has been routed using rules as per greedy channel router algorithm.

Column 7:

a) <u>Make Feasible Top and Bottom Connections in Minimal Manner:</u>

Connect $T_7 = 1$ to track 3 and $B_7 = 7$ to track 4 i.e. track already assigned to this net.

=> Y(2) = {}, Y(3) = {}, Y(1) = {3,4}, Y(6) = {1}, Y(7) = {2,5}

b) <u>Free Up As Many Tracks As Possible By Collapsing Split Nets:</u>

Added pattern of one “collapsing jog” to connect tracks 3 and 4, both holding net 1. This pattern free's up one track

=> Y(3) = {}, Y(2) = {}, Y(1) = {4}, Y(6) = {1}, Y(7) = {2,5}

c) <u>Add Jogs To Reduce The Range of Split Nets:</u>

Although, split nets exist(net 7), jogs cannot be added in this step because it would be incompatible with vertical wiring already placed in this column by step (b).

d) <u>Add Jogs to Raise Rising Nets and Lower Falling Nets:</u>

Net 6 is already closest to its target edge. Jog has already been added for net 1 (now a falling net) to bring it closer to its target edge. Net 7 is a steady net now.

* No need for step (e) i.e. no need to widen channel because T_7 and B_7 were brought in to the channel

f) <u>Extend to next column</u>

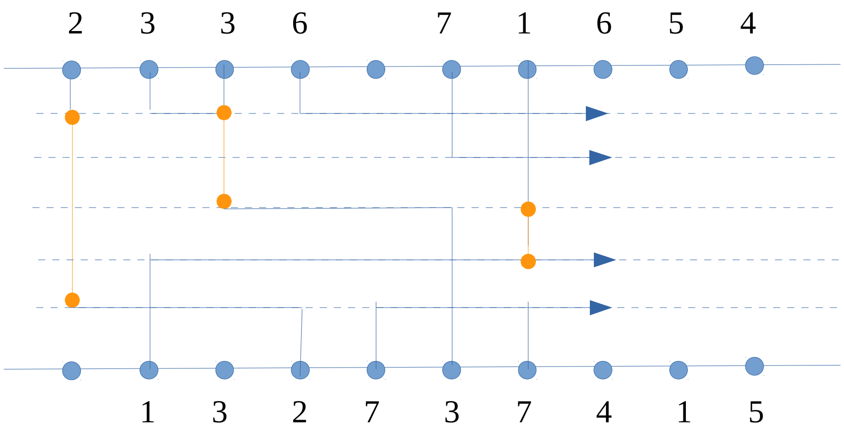

Figure 11: Routing for given routing instance after column 7 has been routed using rules as per greedy channel router algorithm.

Column 8:

a) <u>Make Feasible Top and Bottom Connections in Minimal Manner:</u>

Connect $T_8 = 6$ to track 1 i.e. track already assigned to this net and $B_8 = 4$ to track 3 i.e. to the nearest possible track which is empty. This finishes net 6.

=> Y(2) = {}, Y(3) = {}, Y(1) = {4}, Y(6) = {}, Y(7) = {2,5}, Y(4) = {3} => Net 7 is a split net

b) <u>Free Up As Many Tracks As Possible By Collapsing Split Nets:</u>

Although, split nets exist(net 7), "collapsing jogs" cannot be added in this step because the jog placed for net 7 between tracks 2 and 5 would overlap vertical wiring for net 4 placed in step (a).

c) <u>Add Jogs To Reduce The Range of Split Nets:</u>

Although, split nets exist(net 7), jogs cannot be added in this step because it would be incompatible with vertical wiring already placed in this column by step (a).

d) <u>Add Jogs to Raise Rising Nets and Lower Falling Nets:</u>

Net 4 is already closest to its target edge because tracks 1 and 2 are already occupied. Net 1 (which is a falling net) is already closest to its target edge. Net 7 is a steady net.

* No need for step (e) i.e. no need to widen channel because T_8 and B_8 were brought in to the channel

f) <u>Extend to next column</u>

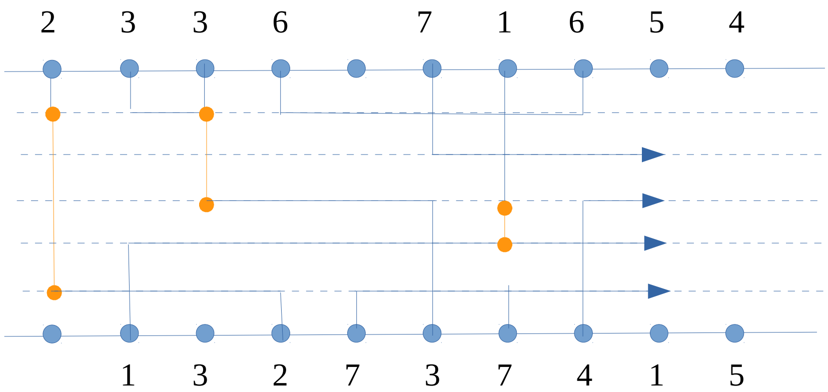

Figure 12: Routing for given routing instance after column 8 has been routed using rules as per greedy channel router algorithm.

Column 9:

a) <u>Make Feasible Top and Bottom Connections in Minimal Manner:</u>

Connect $B_9 = 1$ to track 4 i.e. track already assigned to this net and $T_9 = 5$ to track 1 i.e. to the nearest possible track which is empty. This finishes net 1.

=> Y(2) = {}, Y(3) = {}, Y(1) = {}, Y(6) = {}, Y(7) = {2,5}, Y(4) = {3}, Y(5) = {1}

=> Net 7 is a split net

b) Free Up As Many Tracks As Possible By Collapsing Split Nets:

Although, split nets exist(net 7), "collapsing jogs" cannot be added in this step because the jog placed for net 7 between tracks 2 and 5 would overlap vertical wiring for net 1 placed in step (a).

c) Add Jogs To Reduce The Range of Split Nets:

Although, split nets exist(net 7), jogs cannot be added in this step because it would be incompatible with vertical wiring already placed in this column by step (a).

d) Add Jogs to Raise Rising Nets and Lower Falling Nets:

Net 4 is already closest to its target edge because tracks 1 and 2 are already occupied. Net 7 is a steady net. Net 5 is already closest ot its target edge because the remaining tracks are already occupied.

* No need for step (e) i.e. no need to widen channel because T_9 and B_9 were brought in to the channel

f) Extend to next column

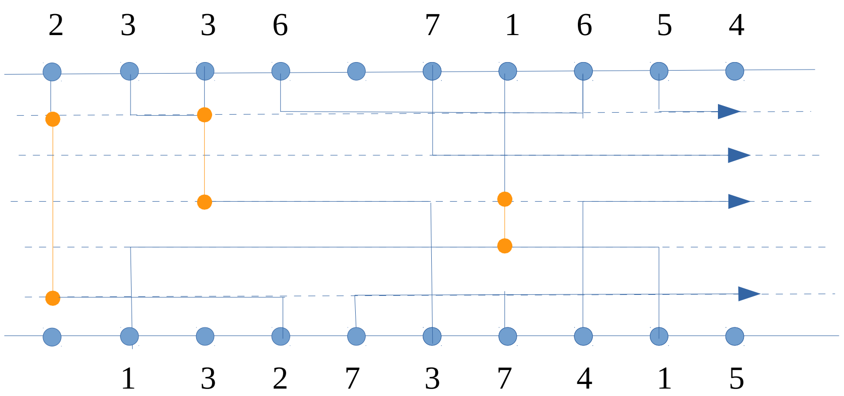

Figure 13: Routing for given routing instance after column 9 has been routed using rules as per greedy channel router algorithm.

Column 10:

a) Make Feasible Top and Bottom Connections in Minimal Manner:

Connect $T_{10} = 4$ to track 3 i.e. track already assigned to this net and $B_{10} = 5$ to track 4 i.e. to the nearest possible track which is empty. This finishes net 4.

=> Y(2) = {}, Y(3) = {}, Y(1) = {}, Y(6) = {}, Y(7) = {2,5}, Y(4) = {}, Y(5) = {1,4}

=> Net 7 and net 5 are split nets

b) Free Up As Many Tracks As Possible By Collapsing Split Nets:

Although, split nets exist(net 7 and net 5), "collapsing jogs" cannot be added in this step because the jog that would be placed for net 7 or net 5 would overlap vertical wiring for net 4 placed in step (a).

c) Add Jogs To Reduce The Range of Split Nets:

Although, split nets exist(net 7 and net 5), jogs cannot be added in this step because it would be incompatible with vertical wiring already placed in this column by step (a).

d) Add Jogs to Raise Rising Nets and Lower Falling Nets:

Net 7 and net 5 are steady nets so need for step (d)

* No need for step (e) i.e. no need to widen channel because T_{10} and B_{10} were brought in to the channel

f) <u>Extend to next column</u>

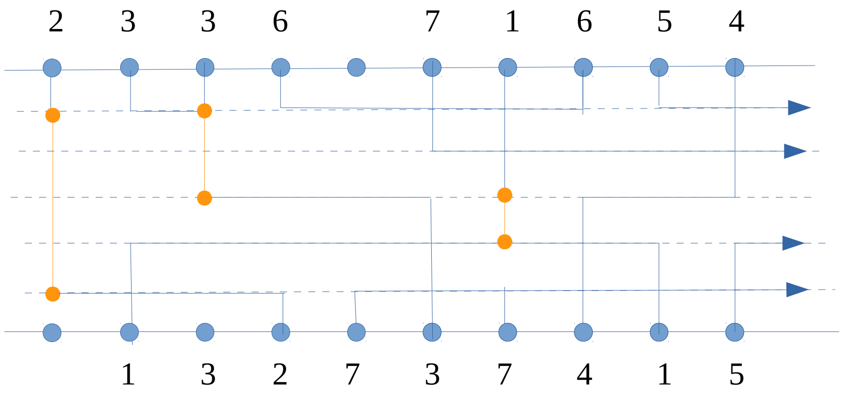

Figure 14: Routing for given routing instance after column 10 has been routed using rules as per greedy channel router algorithm.

Column 11:

* No need for step (a) because there are no terminal pins in column 11.

b) <u>Free Up As Many Tracks As Possible By Collapsing Split Nets:</u>

Place collapsing jog randomly for net 7 because tracks 1 and 4 for net 5 & tracks 2 and 5 for net 7 are symmetric, hence, both nets leave the outermost uncollapsed split net equally far from the channel edge. This completes net 7.

=> Y(2) = {}, Y(3) = {}, Y(1) = {}, Y(6) = {}, Y(7) = {}, Y(4) = {}, Y(5) = {1,4}

=> Net 5 is a split net.

c) <u>Add Jogs To Reduce The Range of Split Nets:</u>

Although, split net exists(net 5), jogs cannot be added in this step because it would be incompatible with vertical wiring already placed in this column by step (b).

d) <u>Add Jogs to Raise Rising Nets and Lower Falling Nets:</u>

Net 5 is a steady net so need for step (d)

* No need for step (e) i.e. no need to widen channel because there are no terminals in column 11.

f) <u>Extend to next column</u>

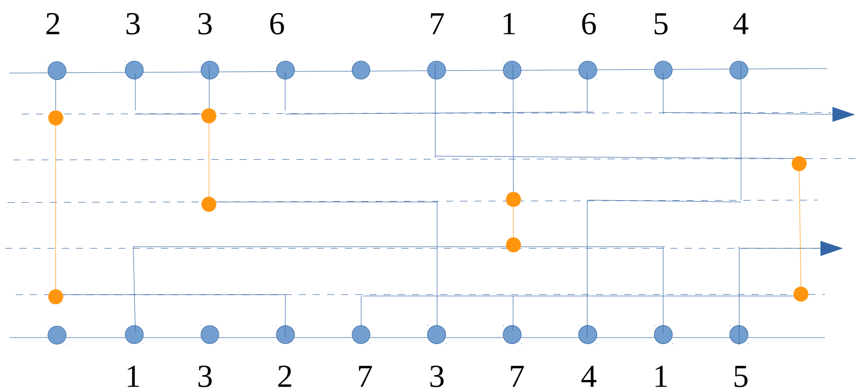

Figure 15: Routing for given routing instance after column 11 has been routed using rules as per greedy channel router algorithm.

Column 12

* No need for step (a) because there are no terminal pins in column 11.
  b) <u>Free Up As Many Tracks As Possible By Collapsing Split Nets:</u>
  Place collapsing jog for net 5. This completes net 5.
  => Y(2) = {}, Y(3) = {}, Y(1) = {}, Y(6) = {}, Y(7) = {}, Y(4) = {}, Y(5) = {}
* No need for step (c) because there are no split nets.
* No need for step (d) because there are no rising or falling nets.
* No need for step (e) i.e. no need to widen channel because there are no terminals in column 12.
* Since, there are no split nets that remain to be collapsed and column 12 ≥ 10, the greedy channel router ends.

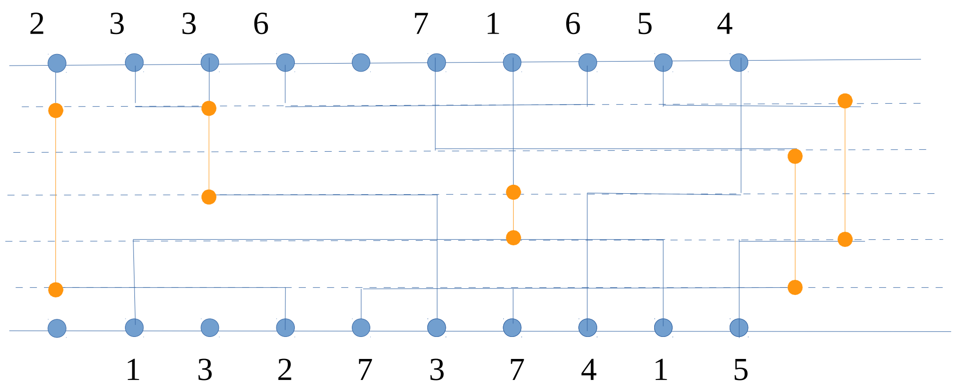

Figure 16: Routing for given routing instance after column 12 has been routed using rules as per greedy channel router algorithm.

This is the final routing Solution obtained.